Sistemas de Arquivos

www.4linux.com.br

Sumário

Índice de tabelas

Índice de Figuras

Capítulo 1

Sistemas de Arquivos

1.1. Objetivos

- Teoria dos sistemas de arquivos.

1.2. Mãos a obra

- **ext2** é antigo e considerado como o filesystem original do Linux, entretanto ele não tem journaling, o que significa que checagens nesse filesystem podem demorar bastante;

- **ext3** é a versão do ext2 com **journaling**, o que proporciona uma recuperação muito mais rápida. O ext3 é um sistema de arquivos muito bom e **confiável**;

- **reiserfs** é um sistema de arquivos com uma performance muito boa e que supera o ext2 e ext3 em termos de desempenho (de 10-15 vezes) quando tratando de pequenos arquivos (**abaixo de 4k**); recomendado para uso geral e para casos mais extremos como o de criação de sistemas de arquivos grandes, utilização com muitos arquivos pequenos ou muito grandes e diretórios com dezenas de milhares de arquivos. Uma desvantagem desse sistema de arquivos a **incapacidade** de utilizá-lo com sistema de quotas e atributos estendidos (em sua versão 3.6);

- **xfs** é um flesystem com **journaling** que possui muitas **funcionalidades** interessantes e é otimizado para escalabilidade. Recomendado para utilização em sistemas com HD SCSI e fibra óptica. Como o xfs faz muito cache dos dados em trânsito na **memória RAM e muitos programas** mal projetados não tomam precauções na hora de escrever arquivos no disco (há muitos programas que fazem isso), se houver uma falha de energia durante escritas em disco podem ocorrer perdas de dados. Sugerimos utilização em sistemas que possuírem no-breaks.

Capítulo 2

Gerenciando

2.1. Objetivos

- Trabalhar com os sistemas de arquivos;
- Conhecer suas diferenças.

2.2. Troubleshooting

Na prática, quando nosso sistema é iniciado o arquivo **/etc/fstab** é lido para que sejam montados as partições em seus pontos de montagens, junto com seus sistemas de arquivos que cada dispositivo está utilizando. Para visualizar o arquivo:

```
# cat /etc/fstab
```

Dentro deste arquivo podemos verificar seis colunas, onde temos o dispositivo, ponto de montagem, sistemas de arquivos, para visualizar as **partições montadas**:

```
# cat /etc/mtab
```

```
# cat /proc/mounts
```

Só que eles não mostram com detalhes o tamanho total e de utilização de cada partição, para isso utilizamos:

```
# df -h
```

Repare que o sistema exibe somente as partições que estão montadas, mas não todas que estão criadas no sistema, uma alternativa para podermos visualizar estas partições:

```
#  cat /proc/partitions
```

Agora iremos fazer um teste, iremos ver como cada **Sistemas de Arquivos** trabalha. Para isso vamos abrir o cfdisk, onde criaremos algumas partições:

```
# cfdisk
```

2.2.1. Cfdisk

Dentro do **cfdisk** vamos criar cinco novas partições, por exemplos, digamos que seja um disco Sata utilizando até a partição sda9:

Partições	Tamanho
/dev/sda10	2 GB
/dev/sda11	2 GB
/dev/sda12	2 GB
/dev/sda13	2 GB
/dev/sda14	2 GB

Repare que os tamanhos estão iguais. Isso é vital para que possamos realizar nossos testes. Preste atenção, pode ser qualquer disco, mais os tamanhos devem ser idênticos.

Para visualizar todas as partições criadas na máquinas:

```
# cfdisk -Ps /dev/sda
```

Mas o fato das partições estarem criadas na máquina, não significa que estão prontas para serem utilizadas, se visualizarmos o arquivo “partitions” iremos perceber que ainda não podemos utiliza-las:

```
# cat /proc/partitions
```

Para que possamos trabalhar com as novas partições é necessário que **reiniciamos** a máquina, e quando ela voltar visualize o arquivo “partitions” e veja que estão prontas :

```
# reboot
```

```
# cat /proc/partitions
```

2.3. Trabalhando com os Sistema de Arquivos

Bom, agora estamos prontos para começar nossos testes. Verifique todos os sistemas de arquivos que nossa máquina dá suporte:

```
# cat /proc/filesystems
```

Podemos verificar que somente as partições que estamos utilizando estão montadas, mas não temos todas, porque iremos utiliza-las agora:

```
# df -h
```

Nosso sistema por padrão já tem o Ext2 e o Ext3 instalados. Verifique se o XFS e ReiserFS:

```
# dpkg -l | grep xfs
```

```
# dpkg -l | grep reiserfs
```

Caso não tenho eles instalados, para instalar vamos utilizar o aptitude:

```
# aptitude install xfsprogs
```

```
# aptitude install reiserfsprogs
```

No nosso caso, o pacote “e2fsprogs” já vem instalado. Ele é responsável pelos programas Ext2 e Ext3:

```
# dpkg  -l  e2fsprogs
```

Verifique os detalhes do pacote:

```
# dpkg -s  e2fsprogs
```

Bom, antes de começarmos efetivamente com os testes, vamos visualizar as partições e o total de Inodes para cada uma. **Inode** nada mais é do que a referencia física do arquivo em disco.

```
# df -h
# df -i
```

Vamos aplicar os Sistemas de arquivos nas partições:

```
# mkfs.xfs /dev/sda11
```

```
# mkfs.reiserfs /dev/sda12
```

```
# mkfs.ext2 /dev/sda13
```

```
# mkfs.ext3 /dev/sda14
```

Lembrando que todas elas estão com **tamanhos iguais**. Vamos criar os pontos de montagem:

```
# mkdir /mnt/xfs
```

```
# mkdir /mnt/reiserfs
```

```
# mkdir /mnt/ext2
```

```
# mkdir /mnt/ext3
```

Vamos montar as partições:

```
# mount -t  xfs  /dev/sda11  /mnt/ xfs
```

```
# mount -t  reiserfs  /dev/sda12  /mnt/reiserfs
```

```
# mount -t   ext2 /dev/sda13 /mnt/ext2
```

```
# mount -t   ext3 /dev/sda14  /mnt/ext3
```

Verifique que todas as partições estão montadas:

```
# df -h
```

Agora verifique a quantidade de inodes, e perceba as diferenças:

```
# df -i
```

Partições	Sistemas de Arquivos	Quantidade em MB	Quantidade de Inodes
/dev/sda11	XFS	4,2M	1951808
/dev/sda12	REISERFS	33M	0
/dev/sda13	EXT2	2,9M	122160
/dev/sda14	EXT3	35M	122160

Como podemos perceber, o Sistemas de Arquivos é um dos menores em ocupação em blocos, e o maior em quantidade de inodes. A Grande diferença do ext2 para o ext3 é o journaling. Quem utiliza o journaling é o xfs e o ext3, em termos de segurança e confiança o ext3 é melhor, sua verificação e restauração são ótimas.

O journaling tem uma tarefa muito importante, ele é responsável por dar a permissão ao Sistema Operacional de que é preciso manter um log das mudanças ocorridas no sistemas de arquivos. Esse log é feito antes de escrever os dados no disco em que está formatado com journaling. Nesse caso do ext3 e do XFS, temos os inodes com as informações do arquivo, já no caso do Reiserfs é diferente, mas também utiliza inodes. O ReiserFS pode ser utilizado no caso de existir uma partição com muitos arquivos pequenos que ocupariam muitos inodes.

Também devemos saber que sistemas de arquivos com journaling tem a possibilidade de oferecer uma maior probabilidade de não se corromper tão facilmente, possibilitando que os dados permaneçam seguros e que não corrompam facilmente quando o sistema travar ou faltar energia, sua recuperação no caso é mais rápida, porque ele não verifica o disco todo, somente verifica a área, partição em que ocorreu o problema, e dentro da partição verifica o log dos arquivos que foram fechados subitamente ou indevidamente.

Além do Ext3, ReiserFS e XFS, encontramos o Ext4, Reiser4, JFS, JFFS, JFFS2 e LogFS. O NTFS também utiliza, mas de uma forma não tão eficaz que feito no GNU/Linux. No caso do Ext4 e do Reiser4, ainda estão em fase de desenvolvimento, mas encontramos nos novo kerneis.